

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-029850

(43)Date of publication of application : 28.01.2000

(51)Int.Cl.

G06F 15/16

(21)Application number : 10-194180

(71)Applicant : NEC DATA TERMINAL LTD

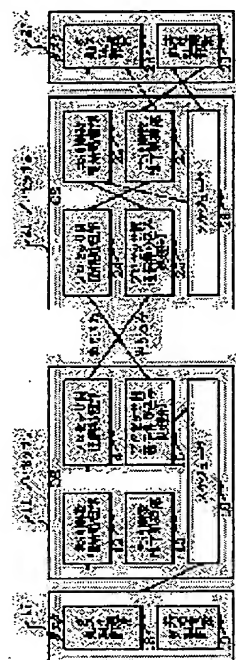
(22)Date of filing : 09.07.1998

(72)Inventor : HARADA KAZUMI

(54) TASK CONTROLLING METHOD USING INTER-PROCESSOR COMMUNICATION OF OPERATING SYSTEM**(57)Abstract:**

PROBLEM TO BE SOLVED: To provide a method capable of allowing an operating system(OS) to be driven between plural processors to execute inter-processor communication by using interruption and inter-processor memory access and control a task of the other processor.

SOLUTION: When a task 17 of a processor 1 requests processing by a system call to a task 27 of a processor 2, an OS 11 in the processor 1 generates an interruption to the processor 2 by an inter-processor communication part 14 and sends a message to an OS 21 in the processor 2. The OS 21 receives the message from the OS 11 by an inter-processor communication interruption processing part 25 and gives processing execution right to the task 27 through a scheduler 26. Thus inter-processor communication is executed between the OSs 11, 21 of respective processors 1, 2 to control the task of the other processor.

**LEGAL STATUS**

[Date of request for examination] 09.07.1998

[Date of sending the examiner's decision of rejection] 08.07.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-29850
(P2000-29850A)

(43) 公開日 平成12年1月28日 (2000.1.28)

(51) Int.Cl.⁷

G 0 6 F 15/16

識別記号

4 3 0

F I

G 0 6 F 15/16

テーマコード(参考)

4 3 0 B 5 B 0 4 5

審査請求 有 請求項の数 6 O L (全 8 頁)

(21) 出願番号 特願平10-194180

(22) 出願日 平成10年7月9日 (1998.7.9)

(71) 出願人 000232025

日本電気データ機器株式会社
東京都調布市上石原3丁目49番地1

(72) 発明者 原田 和美

東京都調布市上石原3丁目49番地1 日本
電気データ機器株式会社内

(74) 代理人 100082935

弁理士 京本 直樹 (外2名)

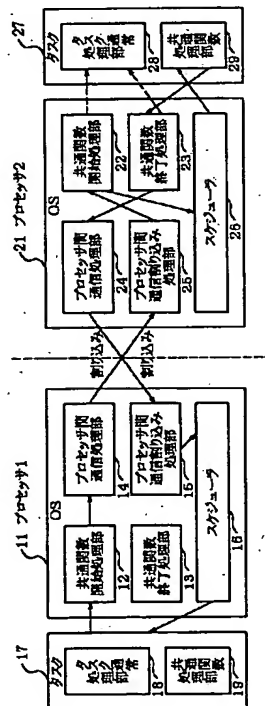
Fターム(参考) 5B045 BB34 BB45 DD10

(54) 【発明の名称】 オペレーティングシステムのプロセッサ間通信を使用したタスク制御方法

(57) 【要約】

【課題】 複数のプロセッサ間で動作するオペレーティングシステム (OS) が割り込みとプロセッサ間メモリアクセスを用いてプロセッサ間通信を行い、他のプロセッサにあるタスクを制御する方法を提供する。

【解決手段】 このオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法は、プロセッサ1のタスク17がプロセッサ2のタスク27に対しシステムコールにより処理を依頼した場合、プロセッサ1のOS 11はプロセッサ間通信処理部14によりプロセッサ2に対し割り込みを発生させ、プロセッサ2のOS 21に対しメッセージを送る。OS 21はプロセッサ間通信割り込み処理部25でOS 11からメッセージを受け取り、スケジューラ26を介して、タスク27に処理の実行権を与える。このようにして、それぞれのプロセッサにあるOS間でプロセッサ間通信を行い、他プロセッサのタスクの制御を可能にする。



【特許請求の範囲】

【請求項1】 複数のプロセッサと、それらのプロセッサがコモンバスを介して、それぞれのプロセッサが他の任意のプロセッサに割り込みを発生させる割り込み処理手段と、それぞれのプロセッサが他の任意のプロセッサのメモリをアクセスすることのできる手段とを備えたマルチプロセッサシステムにおいて、それぞれのプロセッサで動作するオペレーティングシステムが、割り込みとプロセッサ間メモリアクセスとを用いてプロセッサ間通信を行い、他のプロセッサにあるタスクを制御することを特徴とするオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法。

【請求項2】 あるプロセッサ内の各種の処理を行う仕事の単位であるタスクが、他のプロセッサ内のタスクに対しシステムコールにより処理を依頼した場合、前記あるプロセッサ内のオペレーティングシステムは、プロセッサ間通信処理部により前記他のプロセッサに対し割り込みを発生させ、前記他のプロセッサ内のオペレーティングシステムに対しメッセージを送り、前記他のプロセッサのオペレーティングシステムは、プロセッサ間通信割り込み処理部で前記あるプロセッサ内のオペレーティングシステムからメッセージを受け取り、現在実行できる状態にあるタスクの中から一番高いプライオリティのタスクに実行権を与えタスク実行のための準備を行う前記他のプロセッサ内のスケジューラを介して、前記他のプロセッサ内のタスクに処理の実行権を与えることにより、それぞれのプロセッサにあるオペレーティングシステム間でプロセッサ間通信を行い、他のプロセッサのタスクの制御することを特徴とするオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法。

【請求項3】 仕事の単位毎に分割されている処理部であるタスク通常処理部と共通の関数を有する共通関数処理部とから成り各種の処理を行う仕事の単位であるタスクと、前記タスクが有する前記共通関数処理部を動作させるための準備を行う共通関数開始処理部と、前記タスクが有する前記共通関数処理部の動作を終了させ前記タスク通常処理部に戻す準備を行う共通関数終了処理部と、プロセッサ間通信を行うために相手プロセッサのメモリのメッセージ領域にメッセージ内容を書き込むプロセッサ間通信処理部と、前記プロセッサ間通信の割り込みに対する処理を行うプロセッサ間通信割り込み処理部と、現在実行できる状態にある前記タスクの中から一番高いプライオリティの前記タスクに実行権を与えタスク実行のための準備を行うスケジューラとから成る、プロセッサ間通信を制御するオペレーティングシステムと、を有する複数のプロセッサを備えることを特徴とするオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法。

【請求項4】 前記タスクを前記タスク通常処理部と前記共通関数処理部とに区別し、前記共通関数処理部は前

記タスクの所属するプロセッサからでも他のプロセッサからでも実行を依頼できることを特徴とする請求項3記載のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法。

【請求項5】 前記タスクの前記共通関数処理部を実行させるとき、前記タスク通常処理部を中断し、前記共通関数処理部の処理が終了した時点で再開させるオペレーティングシステムを備えることを特徴とする請求項3または4記載のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法。

【請求項6】 共通関数処理の実行依頼をシステムコールでサポートするオペレーティングシステムを備えることを特徴とする請求項3、4または5記載のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はオペレーティングシステム（以下OSと略す）のプロセッサ間通信を使用したタスク制御方法に関する。

【0002】

【従来の技術】 従来、この種のマルチプロセッサシステムのプロセッサ間通信方法は、たとえば特開平3-42762号公報に示されるように、通信先のプロセッサの割り込み処理部でOSに対するシステムコールを発行し、その処理結果を別のプロセッサ間通信によって通信元に通知するようにするものである。

【0003】

【発明が解決しようとする課題】 この従来技術における第1の問題点は、処理依頼を受けたプロセッサは、プロセッサ間割り込み処理部で処理が張り付いてしまうということである。その理由は、プロセッサ間通信割り込み処理部でシステムコールを発行し、その処理結果を依頼元プロセッサに通知するため、処理結果がでるまで待っているためである。

【0004】 第2の問題点は、依頼する処理は、依頼先プロセッサで、OSに対するシステムコールしかないということである。その理由は、プロセッサ間通信の割り込みで処理依頼したとき、プロセッサ間通信割り込み処理部は、システムコールを発行し処理結果を依頼元のプロセッサに通知するのみで、依頼先プロセッサ内のタスクの実行とは無関係であるためである。

【0005】 本発明の目的は、複数のプロセッサ間で動作するOSが割り込みとプロセッサ間メモリアクセスを用いてプロセッサ間通信を行い、他のプロセッサにあるタスクを制御する方法を提供することにある。

【0006】

【課題を解決するための手段】 本発明のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法は、複数のプロセッサと、それらのプロセッサがコモ

3

ンバスを介して、それぞれのプロセッサが他の任意のプロセッサに割り込みを発生させる割り込み処理手段と、それぞれのプロセッサが他の任意のプロセッサのメモリをアクセスすることのできる手段とを備えたマルチプロセッサシステムにおいて、それぞれのプロセッサで動作するオペレーティングシステムが、割り込みとプロセッサ間メモリアクセスとを用いてプロセッサ間通信を行い、他のプロセッサにあるタスクを制御することを特徴とする。

【0007】また、本発明のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法は、あるプロセッサ内の各種の処理を行う仕事の単位であるタスクが、他のプロセッサ内のタスクに対しシステムコールにより処理を依頼した場合、前記あるプロセッサ内のオペレーティングシステムは、プロセッサ間通信処理部により前記他のプロセッサに対し割り込みを発生させ、前記他のプロセッサ内のオペレーティングシステムに対しメッセージを送り、前記他のプロセッサのオペレーティングシステムは、プロセッサ間通信割り込み処理部で前記あるプロセッサ内のオペレーティングシステムからメッセージを受け取り、現在実行できる状態にあるタスクの中から一番高いプライオリティのタスクに実行権を与えタスク実行のための準備を行う前記他のプロセッサ内のスケジューラを介して、前記他のプロセッサ内のタスクに処理の実行権を与えることにより、それぞれのプロセッサにあるオペレーティングシステム間でプロセッサ間通信を行い、他のプロセッサのタスクの制御することを特徴とする。

【0008】さらに、本発明のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法は、仕事の単位毎に分割されている処理部であるタスク通常処理部と共通の関数を有する共通関数処理部とから成り各種の処理を行う仕事の単位であるタスクと、前記タスクが有する前記共通関数処理部を動作させるための準備を行う共通関数開始処理部と、前記タスクが有する前記共通関数処理部の動作を終了させ前記タスク通常処理部に戻す準備を行う共通関数終了処理部と、プロセッサ間通信を行うために相手プロセッサのメモリのメッセージ領域にメッセージ内容を書き込むプロセッサ間通信処理部と、前記プロセッサ間通信の割り込みに対する処理を行うプロセッサ間通信割り込み処理部と、現在実行できる状態にある前記タスクの中から一番高いプライオリティの前記タスクに実行権を与えタスク実行のための準備を行うスケジューラとから成る、プロセッサ間通信を制御するオペレーティングシステムと、を有する複数のプロセッサを備えることを特徴とする。

【0009】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して詳細に説明する。

【0010】図1は、本発明の一実施の形態のファーム

4

ウェア機能を示すブロック図である。図1を参照すると、この実施の形態のオペレーティングシステムのプロセッサ間通信を使用したタスク制御方法は、プロセッサ1とプロセッサ2とから構成され、それぞれのプロセッサ1、2は、タスク17およびタスク27とタスク17、27を制御するOS11およびOS21とを含む。OS11およびOS21は、複数タスクを制御するが、この実施の形態では、タスクを単数で記載する。

【0011】プロセッサ1のOS11は、共通関数開始処理部12と、共通関数終了処理部13と、プロセッサ間通信処理部14と、プロセッサ間通信割り込み処理部15と、スケジューラ16とを備えている。また、プロセッサ2のOS21は、プロセッサ1のOS11と同じ構成である。プロセッサ1のタスク17は、タスク通常処理部18と共通関数処理部19とを備えている。プロセッサ2のタスク28も、プロセッサ1のタスク17と同じ構成である。

【0012】タスク17の共通関数処理部19およびタスク27の共通関数処理部29は、他のプロセッサあるいは同一のプロセッサから処理依頼を受けたときに処理を行う。プロセッサ1のタスク通常処理部18で動作しているタスク17に対し、処理依頼がある場合、OS11の共通関数開始処理部12は、タスク通常処理部18を中断し、共通関数処理部19を実行させる。共通関数処理部19は、共通関数処理が終了すると、OS11に終了を通知し、共通関数終了処理部13が、タスク通常処理部18を再開させる。

【0013】共通関数開始処理部12および共通関数開始処理部22は、2つの機能を実現する。1つは、同一プロセッサ内の共通関数処理部を有するタスクに対し、共通関数処理の処理依頼があった場合、同一プロセッサ内のタスクの共通関数を開始するための処理を行う機能を有し、もう1つは、他の任意のプロセッサにある共通関数処理部を有するタスクに対し共通関数処理の処理依頼があった場合、依頼を受けたタスクを待ち状態にし、プロセッサ間通信を行い他のプロセッサに処理依頼をするために、プロセッサ間通信処理部14およびプロセッサ間通信処理部24に制御を移す機能である。

【0014】図2はこの実施の形態にハードウェアの構成を示すブロック図である。図2を参照すると、図1におけるプロセッサ1である51の通信制御コントローラ33は、コモンバス39と、図1におけるプロセッサ2である52の通信制御コントローラ37を介して、割り込みコントローラ36に割り込みを発生させ、メモリ38をアクセスすることができる手段を含み、また、通信制御コントローラ37は、コモンバス39と、通信制御コントローラ33を介して、割り込みコントローラ32に割り込みを発生させ、メモリ34をアクセスすることができる手段を含むことにより、OS11とOS21のプロセッサ間通信を可能とする。

【0015】プロセッサ1のOS11がプロセッサ2のOS21とプロセッサ間通信を行う場合、OS11のプロセッサ間通信処理部14が、プロセッサ2に割り込みを発生させ、OS21のプロセッサ間通信割り込み処理部25がOS11からのプロセッサ間通信を受信する。

【0016】次に、図1に併せて図3、図4、図5の流れ図を参照してこの実施の形態の全体の動作について詳細に説明する。

【0017】特に、図1のプロセッサ1のタスク通常処理部18で動作中のタスク17が、プロセッサ2のタスク27にある共通関数処理部29の処理を依頼する場合の動作を説明する。

【0018】まず、プロセッサ1のタスク17のタスク通常処理部18が、OS11に対し、プロセッサ2のタスク27にある共通関数処理部29の処理をシステムコールで依頼する。OS11の共通関数開始処理部12は、タスク通常処理部18からの処理依頼を受け付けると、タスク17の状態を実行状態から待ち状態に移行する(ステップ101、102)。共通関数開始処理部12は、タスク通常処理部18からの処理依頼の依頼先が同一プロセッサか、他のプロセッサかを判断する(ステップ103)。依頼先が他のプロセッサであるプロセッサ2であるため、プロセッサ間通信処理部14が依頼先プロセッサであるプロセッサ2のメモリのメッセージ領域に依頼内容を通知し(ステップ104)、プロセッサ2に対しプロセッサ間通信の割り込みを発生させる(ステップ105)。

【0019】OS11は、プロセッサ2のOS21から、処理結果のプロセッサ間通信の割り込みが来るまで、別タスクの処理を行うため、スケジューラ16を呼び(ステップ110)、次に実行できるタスクに実行権を与える。

【0020】一方、依頼先であるプロセッサ2のOS21は、プロセッサ1からのプロセッサ間通信の割り込みを受信すると(ステップ201)、プロセッサ間通信割り込み処理部25は、OS11のプロセッサ間通信割り込み処理部15がメッセージを書き込んだプロセッサ2のメモリのメッセージ領域からメッセージを読みとる(ステップ202)。プロセッサ間通信割り込み処理部25は、読み込んだメッセージ内容は共通関数の処理依頼か共通関数の処理結果かを判断し(ステップ203)、メッセージ内容が共通関数の処理依頼であるため、共通関数開始処理部22が、共通関数を開始するための処理を行う。まず、依頼されている共通関数処理部29を有するタスク27のタスク通常処理部28を中断し(ステップ206)、タスク通常処理部28の現在の状態を保存する(ステップ207)。さらに、タスク27の実行アドレスをタスク通常処理部28から共通関数処理部29に変更し(ステップ208)、タスク27のタスクスタックをタスク通常処理部28が動作するため

のスタックから共通関数処理専用のスタックに変更する(ステップ209)。最後にOS21はスケジューラ26を呼び(ステップ210)、タスク27をスケジューリングし、タスク27のプライオリティが実行できる状態のタスクの中で一番高い場合、タスク27に実行権を与え、共通関数処理部29が動作する。

【0021】タスク27の共通関数処理部29の実行が終了すると、共通関数処理部29は、OS21に対し、共通関数の終了と処理結果を通知する。OS21が共通関数の終了通知を受け付けると(ステップ301)、共通関数終了処理部23は、終了通知を発行した共通関数処理部29を有するタスク27に、共通関数処理部29を実行する前に保存した値を復帰させ(ステップ302)、共通関数処理部29を実行するために中断状態にあるタスク通常処理部28の中断状態を解除し、再開させる(ステップ303)。次に、共通関数終了処理部23は、共通関数処理依頼の依頼元が同一のプロセッサか他のプロセッサかを調査し(ステップ304)、依頼元は他のプロセッサであるプロセッサ1であるため、プロセッサ間通信処理部24が、依頼元であるプロセッサ1のメモリのメッセージ領域に処理結果を通知し(ステップ307)、プロセッサ1に対しプロセッサ間通信の割り込みを発生させる(ステップ308)。OS21は、スケジューラ2を呼び(ステップ309)、これでプロセッサ2での共通関数処理依頼に対する処理は終了する。

【0022】プロセッサ2のOS21から共通関数処理依頼に対する処理終了のプロセッサ間通信の割り込みを受信したプロセッサ1のOS11(ステップ201)のプロセッサ間通信割り込み処理部15は、OS21のプロセッサ間通信割り込み処理部24が処理結果のメッセージを書き込んだプロセッサ1のメモリのメッセージ領域からメッセージを読みとる(ステップ202)。プロセッサ間通信割り込み処理部15は、読み込んだメッセージ内容が共通関数処理依頼か共通関数処理結果かを判断し(ステップ203)、メッセージ内容が共通関数処理結果であるため、共通関数処理依頼を受けたタスクに対し、処理結果を通知するための処理を行う。プロセッサ間通信割り込み処理部15は、待ち状態にある処理依頼を受けたタスク17の待ち状態を解除し(ステップ204)、処理依頼を受けたタスク17に対し、処理結果をシステムコールの結果として通知し(ステップ205)、スケジューラを呼ぶ(ステップ210)。

【0023】次に、本発明の他の実施の形態について図面を参照して詳細に説明する。

【0024】前述の実施の形態で述べた2個のプロセッサをN個のプロセッサに拡張した場合を説明する。図6を参照すると、この実施の形態は、N個目のプロセッサをプロセッサ3とし、プロセッサ3が、図1に示されたプロセッサ2に加え、プロセッサ1とのプロセッサ間通

信制御を行う。プロセッサ3のOS41は、プロセッサ1のOS11およびプロセッサ2のOS21と同じ構成であり、プロセッサ3のタスク47は、プロセッサ1のタスク17およびプロセッサ2のタスク27と同じ構成である。

【0025】また、図7を参照すると、プロセッサ1を51、プロセッサ2を52、プロセッサNを53としてみることができ、プロセッサ1である51の通信制御コントローラ33は、コモンバス39と、プロセッサ2である52の通信制御コントローラ37を介して、割り込みコントローラ36に割り込みを発生させ、メモリ38をアクセスすることができるのと同時に、コモンバス39と、プロセッサ3である53の通信制御コントローラ56を介して、割り込みコントローラ55に割り込みを発生させ、メモリ57をアクセスすることができる。

【0026】これにより、プロセッサ1とプロセッサ3の間でのプロセッサ間通信が可能であり、前述の実施の形態で述べたプロセッサ1とプロセッサ2での処理と同様な処理が可能である。

【0027】次に、別の実施の形態として、共通関数処理を同時に複数個依頼する方法を説明する。図6のタスク17はタスク27が所有する共通関数処理部29と、タスク47が所有する共通関数処理部49を同時に依頼する。OS11のプロセッサ間通信処理部14は、プロセッサ2のメモリのメッセージ領域に処理依頼を通知し、同時にプロセッサ3のメモリのメッセージ領域に処理依頼を通知する。次に、プロセッサ間通信処理部14は、プロセッサ2およびプロセッサ3に対し、プロセッサ間通信割り込みを行う。共通関数処理部がOSに対し、処理結果と終了を通知をし、共通関数終了処理部が共通関数処理部からタスク通常処理部への切り替えをするまでの処理は、前述の実施の形態で述べたものと同様の処理である。プロセッサ間通信処理部24およびプロセッサ間通信処理部44は、依頼元であるプロセッサ1のメモリのメッセージ領域にそれぞれ処理結果を通知し、プロセッサ1に対し、プロセッサ間通信の割り込みを行う。プロセッサ1のプロセッサ間通信割り込み処理部15は、プロセッサ2のOS21からの割り込みと、プロセッサ3のOS41からの割り込みを受信し、依頼を受け取ったタスク17に対し、処理結果を通知する。

【0028】

【発明の効果】以上説明したように、本発明の第1の効果は、タスクが、OSにシステムコールを発行し、他のプロセッサの処理を依頼することで、他のプロセッサにあるタスクの処理をあたかも同一プロセッサにある処理の如く実行することができることにある。その理由は、OSがシステムコール処理内でプロセッサ間通信を行

い、他のプロセッサ内にあるOSがそれに応じてタスクの制御を行うためである。

【0029】第2の効果は、処理を依頼した側のプロセッサでは、依頼先のプロセッサでの処理を待つ間、張り付くことなしに、別処理ができることにある。その理由は、OSが依頼を受けたタスクの状態を制御し、プロセッサ間通信の後は、スケジューリングを行っているためである。

【0030】第3の効果は、処理を依頼された側のプロセッサでは、プロセッサ間通信割り込み処理にOSが張り付くことなく、タスクの共通関数を実行することができることにある。その理由は、タスクの通常処理部を中断して共通関数処理部を動作させることにより、共通関数はタスクとして、OSのスケジューリングにより実行権を得ることができるためである。

【0031】第4の効果は、タスクが通常処理部と区別して、他のタスクから処理依頼があった場合に動作する共通関数処理部を持つことにより、あらゆるタスクが共通関数処理部を自分の関数の如く使用できることにある。その理由は、タスクの共通関数コールをOSがシステムコールとしてサポートしているためである。

【図面の簡単な説明】

【図1】本発明の一実施の形態のファームウェア機能構成を示すブロック図である。

【図2】この実施の形態のプロセッサ間通信を行う場合のハードウェア構成を示すブロック図である。

【図3】この実施の形態における動作の一例を示す流れ図である。

【図4】この実施の形態における動作の一例を示す流れ図である。

【図5】この実施の形態における動作の一例を示す流れ図である。

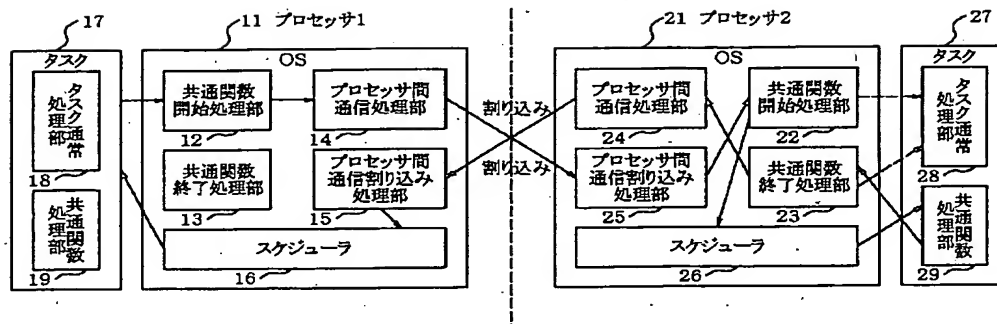
【図6】本発明の他の実施の形態のファームウェア機能構成を示すブロック図である。

【図7】他の実施の形態のプロセッサ間通信を行う場合のハードウェア構成を示すブロック図である。

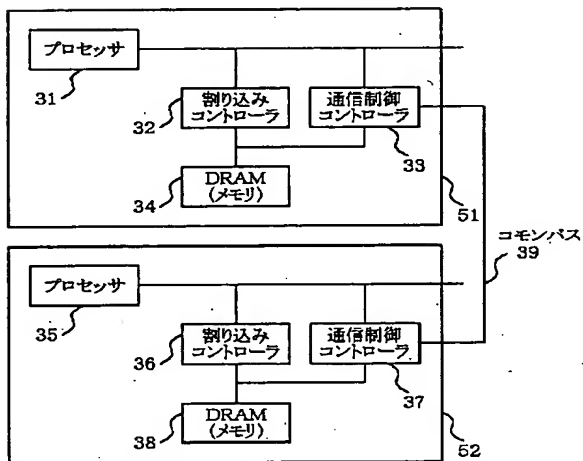
【符号の説明】

1, 2, 3	プロセッサ
11, 21, 41	OS
12, 22, 42	共通関数開始処理部
13, 23, 43	共通関数終了処理部
14, 24, 44	プロセッサ間通信処理部
15, 25, 45	プロセッサ間通信割り込み処理部
16, 26, 46	スケジューラ
17, 27, 47	タスク
18, 28, 48	タスク通常処理部
19, 29, 49	共通関数処理部

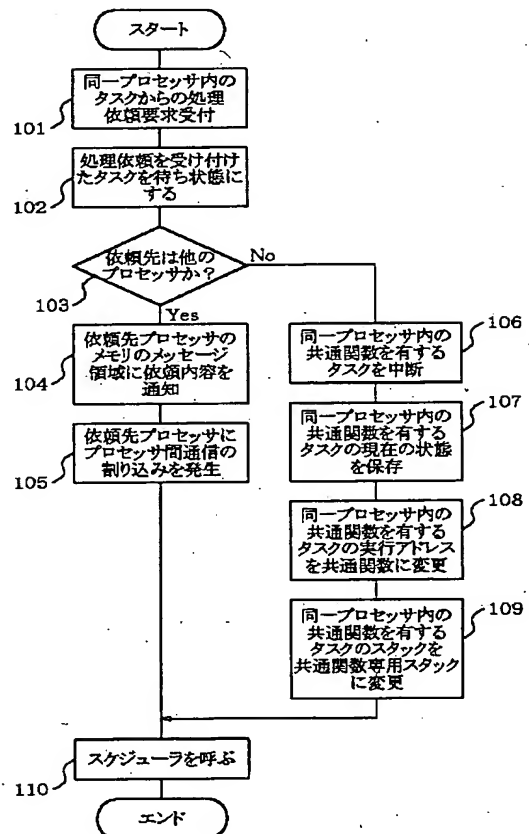
【図1】



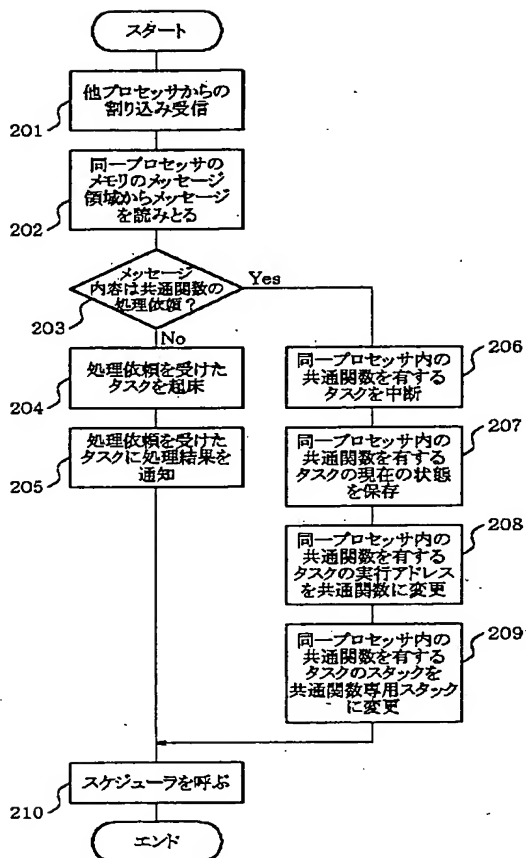
【図2】



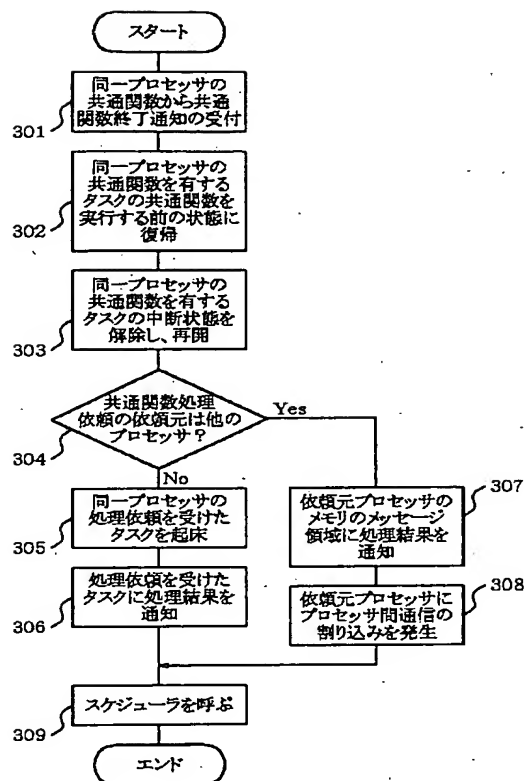
【図3】



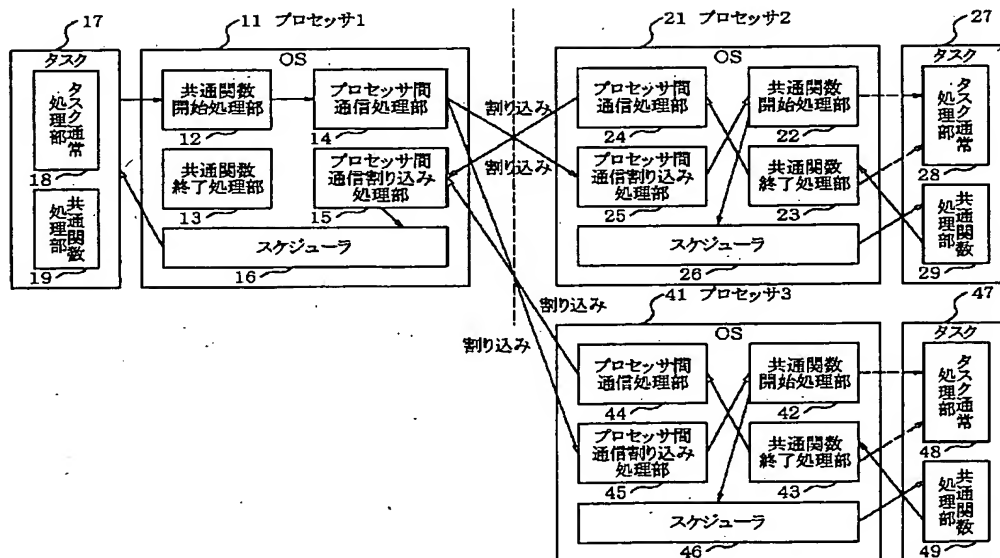
【図4】



【図5】



【図6】



【図7】

